HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

BACHELORARBEIT

Frau
Xinyue Lin

Programmierung eines wissenschaftlichen Taschenrechners auf zwei Multifunktionsboards

2015

Fakultät: Elektro- und Informationstechnik

BACHELORARBEIT

Programmierung eines wissenschaftlichen Taschenrechners auf zwei Multifunktionsboards

Autorin:
Frau Xinyue Lin

Studiengang:
Elektro- und Informationstechnik

Seminargruppe:
EI13w1-BC

Erstprüfer:
Prof. Dr.-Ing. Alexander Lampe

Zweitprüfer:
Markus Süß M.Sc.

Einreichung:
Hochschule Mittweida
2015

Faculty of Electrical Engineering and Information Technology

BACHELOR THESIS

Programming a scientific calculator on two multifunction boards

author:
Ms. Xinyue Lin

course of studies:
Electrical Engineering and Information Technology

seminar group:
EI13w1-BC

first examiner:
Prof. Dr. A. Lampe

second examiner:
Markus Süß M. Sc.

Submission:
Hochschule Mittweida
2015

Bibliografische Angaben

Lin, Xinyue:

Programmierung eines wissenschaftlichen Taschenrechners auf zwei Multifunktionsboards

Programming a scientific calculator on two multifunction boards

41 Seiten, Hochschule Mittweida, University of Applied Sciences, Fakultät Elektro- und Informationstechnik, Bachelorarbeit, 2015

Referat

Das Ziel der Bachelorarbeit ist es, auf zwei Multifunktionsboards, jedes bestehend u.a. aus einem 8-bit Mikrokontroller, 4x4 Tastenmatrix, LCD-Modul 2x8, einen wissenschaftlichen Taschenrechner zu realisieren. Dabei dient ein Multifunktionsboard der Ein und Ausgabe, das andere den Berechnungen.

Alle Programme sind in der Programmiersprache C geschrieben.

Inhaltsverzeichnis

Abkürzungsverzeichnis

TWI	Two-Wire Serial Interface
LCD	Liquid Crystal Display
GND	Ground
SDA	Serial Data Line
SCL	Serial Clock Line
JTAG	Joint Test Action Group
I²C	Inter-Integrated Circuit
RISC	Reduced Instruction Set Computer
SREG	Statusregister
µC	Mikrocontroller
I/O	Input and Output
ISR	Interrupt Service Routine
ACK	Acknowledgement
TWCR	TWI Control Register
TWSR	TWI Status Register
TWDR	TWI Data Register

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Moderne Systeme zur digitalen Signalverarbeitung sind aus denselben Grundkomponenten aufgebaut, wie z.B. Mikrokontroller, Datenein- und -ausgabe-bausteine sowie Speicherelemente. Um auf einer gegebenen Hardware eine gewünschte Signalverarbeitungsaufgabe zu realisieren wird Software sowohl für Grundfunktionen als auch anwendungsspezifische Funktionen benötigt.

Das Thema der Bachelorarbeit ist “Programmierung eines wissenschaftlichen Taschenrechners auf zwei Multifunktionsboards“. Wegen des Themas müssen folgende Aufgaben gelöst werden:

1. Zuerst werden die Funktionen eines wissenschaftlichen Taschenrechners auf den verschiedenen Hardware deutlich eingeteilt, indem man die Aufgabenstellung analysiert:

   a) Die Zahlen und Operatoren werden einem Multifunktionsboard eingegeben, d.h. dieses Multifunktionsboard dient als Master-Board;

   b) Der Mikrocontroller anderes Multifunktionsboards berechnet die Gleichung, d.h. das andere Multifunktionsboard dient als Slave-Board.

   c) Auf dem LCD werden die Eingabe und Ergebnisse angezeigt.

Da ein wissenschaftlicher Taschenrechner zwei Multifunktionsboards besteht, wird ein TWI-Bus für die Kommunikation zwischen beiden Boards genutzt. Für dieses Bus-System existieren aus Lizenzgründen zwei verschiedene Bezeichnungen: TWI und I^2C. Beide werden in dieser Arbeit synonym verwendet.

Die zwei verwendeten Multifunktionsboards bestehen jeweils u.a. aus einem 8-bit Mikrokontroller, 16 Tasten Ein-Ausgabematrix, LCD-Modul 2x8 Zeichen. Das Master-Board soll als Eingabeeinheit wirken und das Slave-Board ist ausschließlich für die Berechnung des Ergebnisses zuständig und sendet dieses wieder zum Master.

Im Einzelnen bedeutet dies:

- Erstellen der Software-Architektur, insbesondere Wahl eines geeigneten Bussystems zur Kopplung der beiden Boards, für die zu realisierende Anwendung inklusive Testkonzept.
- Implementieren der Grundfunktionen zur Steuerung des Systems sowie zur Ein- und Ausgabe in C.
- Implementieren der Grundrechenarten (add, sub, mul, div) sowie spezieller mathematischer Funktionen (cos, sin, exp, ln).
- Test der erstellten Software.
- Dokumentation der Ergebnisse.

2 Aufbau der Hardware

Ein wissenschaftlicher Taschenrechner besteht aus einem LCD-I^{2}C-Modul 2x8 und zwei Multifunktionsboards, die mit der Spannungsversorgung und notwendigen Verdrahtung im Praktikum entworfen und gelötet wurden.

2.1 Gerätebau eines Multifunktionsboards

Die wichtigen Komponenten auf dem ersten Multifunktionsboard(sieht *Abbildung 1*) sind ein Mikrocontroller ATmega32, eine 4×4 Tastenmatrix für die Eingabe und ein LCD-Display für die Ausgabe. Auf dem zweiten Multifunktionsboard benutzt man nur den Mikrocontroller, um die Berechnungen zu implementieren. Die Spannungsversorgung erfolgt mit einer 9V Batterie.

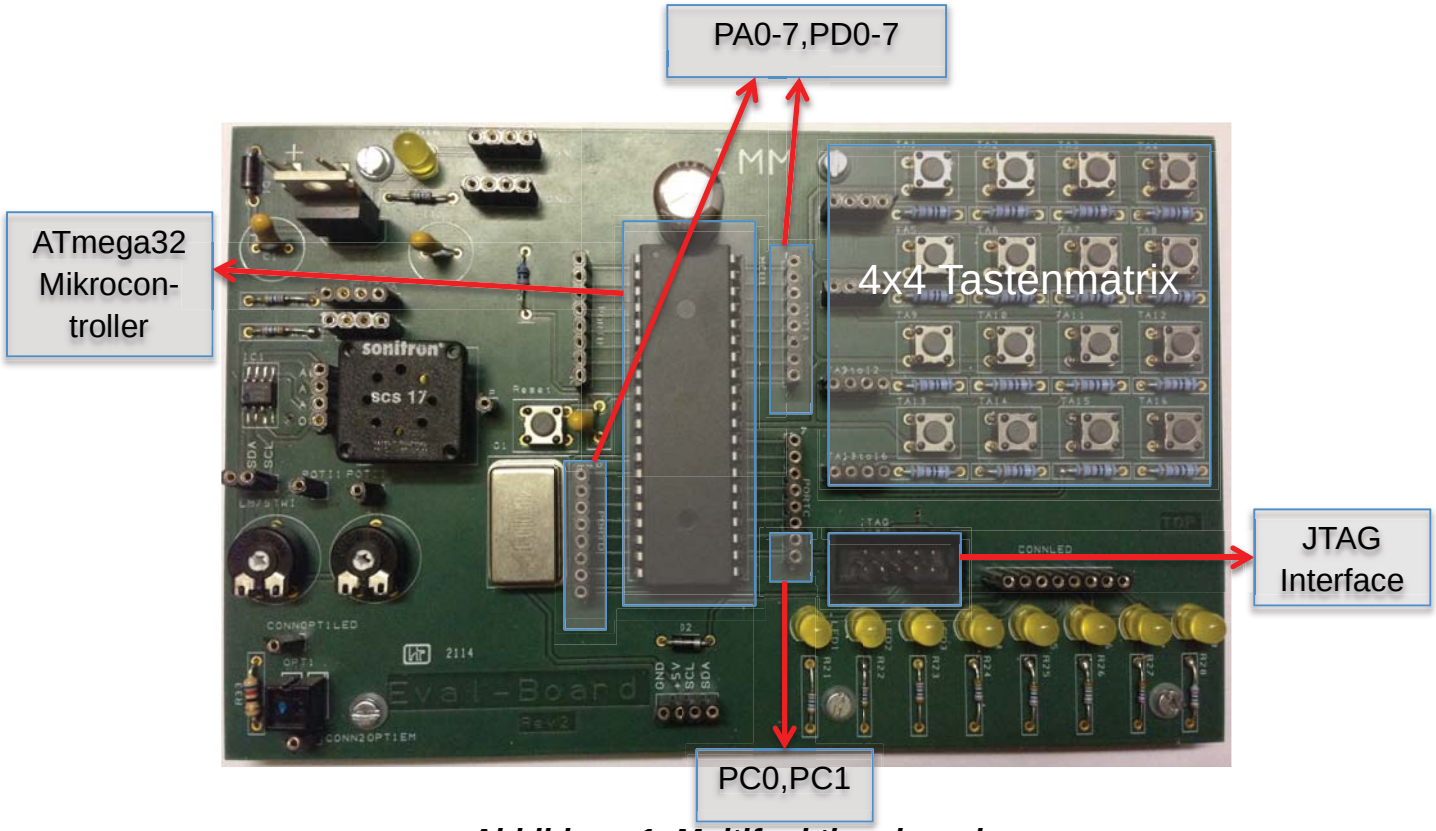

Abbildung 1: Multifunktionsboard

2.1.1 ATmega32

ATmega32 ist ein 8-bit AVR Mikrocontroller mit 32K Bytes In-System programmierbarem Flash-Speicher. Er besitzt eine RISC Architektur und enthält 131 Befehlsätze, meistens Single-Clock Cycle Durchführung. Deswegen er-

langt der Durchsatz annähernd 1 MIPS pro MHz. Außerdem gibt es im ATmega32 non-volatile Programm- und Datenspeicher, 32 8-bit Arbeitsregister, JTAG Interface, 32 programmierbare I/O Ports usw.

2.1.2 JTAG Interface

JTAG Interface dient als den Kommunikationsinterface zwischen Mikrocontroller und JTAG-Emulator. Durch das JTAG Interface und den JTAG-Emulator kann man mittels der Software „AVR Studio 4.0“ Maschinencode auf den Controller überspielen und mit dem Debugger schrittweise ausführen. Außerdem hat man Zugriff auf alle Register und internen Statusvariablen.

2.1.3 4×4 Tastenmatrix

Die verschiedenen Funktionen werden an sechzehn Tasten auf dem Master-Board verteilt. Manche Tasten haben einzelne Funktion, z.B. zehn Zahlen(von 0 bis 9), Komma und das Gleichheitszeichen. Und jede der andren vier Tasten enthält zwei Funktionen. Wenn man falsch eingibt und LCD löschen möchte, kann man mit der Taste „Reset“ drücken. *Abbildung 2* zeigt die Funktionen auf jeder Taste.

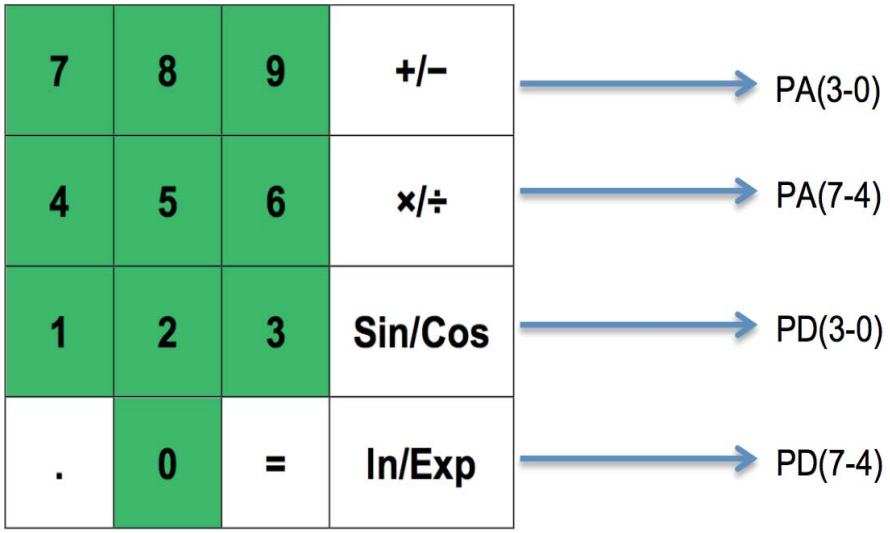

Abbildung 2: Funktionen der 4×4 Tastenmatrix

Wenn eine Taste gedrückt wird, liest µC des Masters den Zustand der Tasten und sendet dem Slave-Board die Eingabe, die zur gleichen Zeit auf der LCD gezeigt wird. Nachdem einer kompletten Gleichung empfangen geworden ist, berechnet das

Slave-Board die Gleichung und dann sendet das Ergebnis zurück. Sobald das Master-Board das Ergebnis empfängt, wird LCD gelöscht und dann zeigt es.

2.2 LCD-I^{2}C-Modul und JTAG Emulator

Außer den beiden Multifunktionsboards werden ein LCD-I^{2}C Adaptermodul und ein JTAG Emulator verwendet. Das LCD-I^{2}C Adaptermodul mit dem Chip „PCF8574P“ schließt LCD-Displays am I^{2}C-Bus an. Das LCD-Display kann zwei Zeilen, maximal 16 Schriftzeichen zeigen. Auf dem LCD werden die Eingaben und die Ergebnisse angezeigt.

Der JTAG-Emulator ist ähnlich wie eine Brücke zwischen Computer und Hardware, denn er hat Zugriff drauf: alle innere und periphere Einheiten, interne und externe RAM, die Datei der interne Register, EEPROM und Flash-Speicher usw. Deswegen können Funktionstest und Programmierung auf der Hardware mit dem JTAG-Emulator durchgeführt werden. Computer und JTAG-Emulator werden mit einem USB-Kabel verbunden und zwischen dem JTAG-Emulator und Hardware ist eine JTAG Schnittstelle.[2]

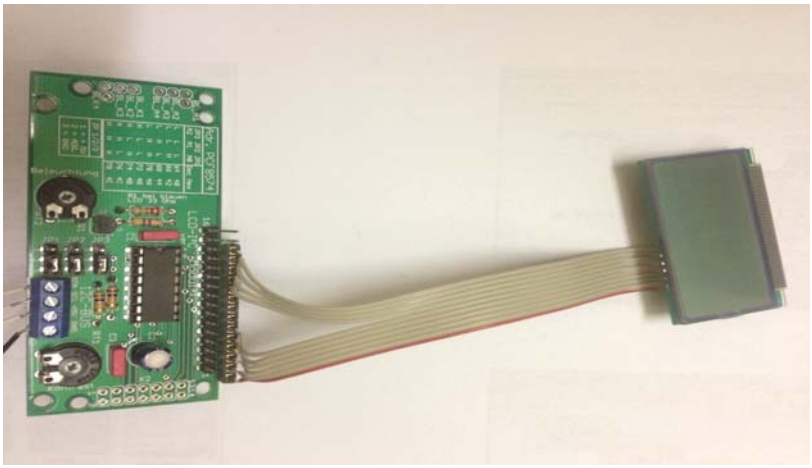

Abbildung 3: LCD-I2C-Modul

2.3 Architektur des Taschenrechners

Beide Multifunktionsboards haben verschiedene Funktionen. Auf dem Master-Board wird die 4×4 Tastenmatrix(TA1-TA16) als Eingabeeinheit benutzt, deshalb dienen Port A und Port D als I/O Ports, um den Zustand der Tasten zu lesen. Dazu werden die Anschlüsse PA3~0 mit TA1~4, PA7~4 mit TA5~8, PD3~0 mit TA9~12 und PD7~4

mit TA13~16 miteinander verbunden. Und um die Kommunikation zwischen beiden Multifunktionsboards herzustellen, koppelt man PC0 und PC1 für die TWI-Verbindung auf beiden Boards. Anschließend wird LCD-I^{2}C-Modul mit dem Master-Board durch vier Leitungen verbunden. Am Ende werden mit zwei Leitungen VCC und GND auf beiden Boards kurzgeschlossen, um die Spannungsversorgung herzustellen. *Abbildung 4* zeigt die Architektur der Hardware.

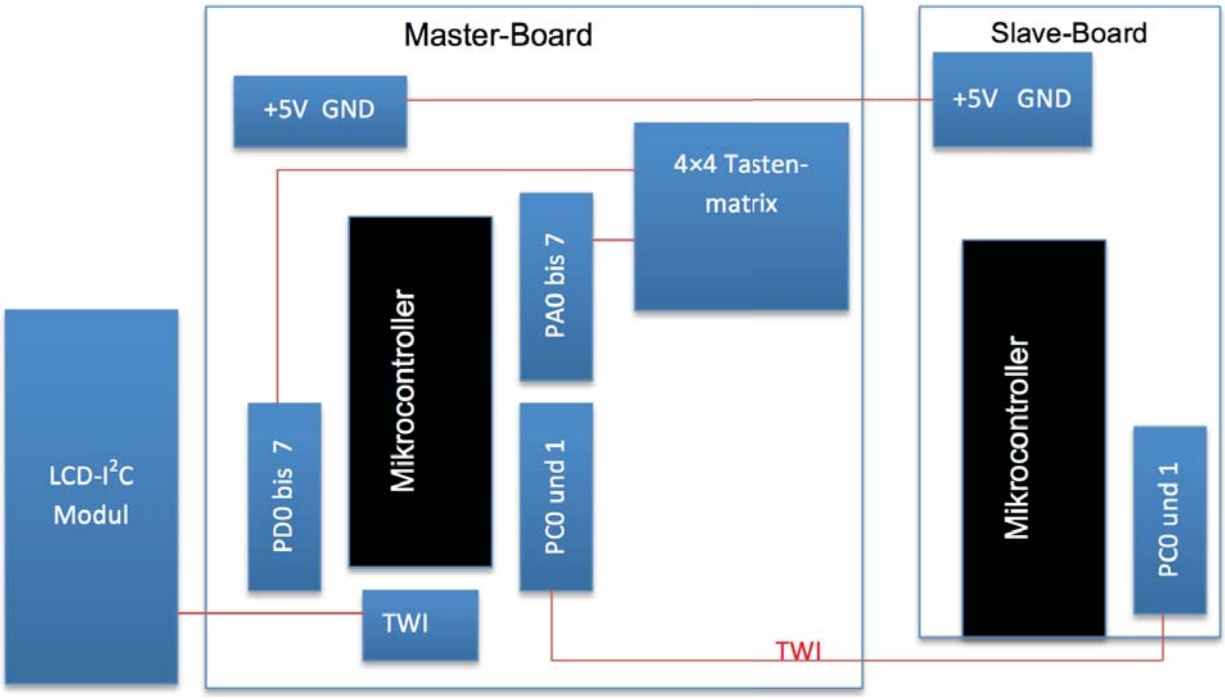

Abbildung 4: Architektur des Taschenrechners

3 Kommunikationsschnittstelle

In diesem Kapitel soll betrachtet werden, wie die Kommunikation zwischen Master und Slave im Detail funktioniert. Dazu werden die Übertragung der Eingaben und der Ergebnisse auf den beiden Boards durchgeführt.

3.1 Kommunikation im Programmablauf

Die Kommunikation zwischen beiden Multifunktionsboards verwendet TWI-Bus, das aus zwei Leitungen besteht. *Abbildung 5* zeigt die wann im Programmablauf Daten über den TWI-Bus gesendet werden.

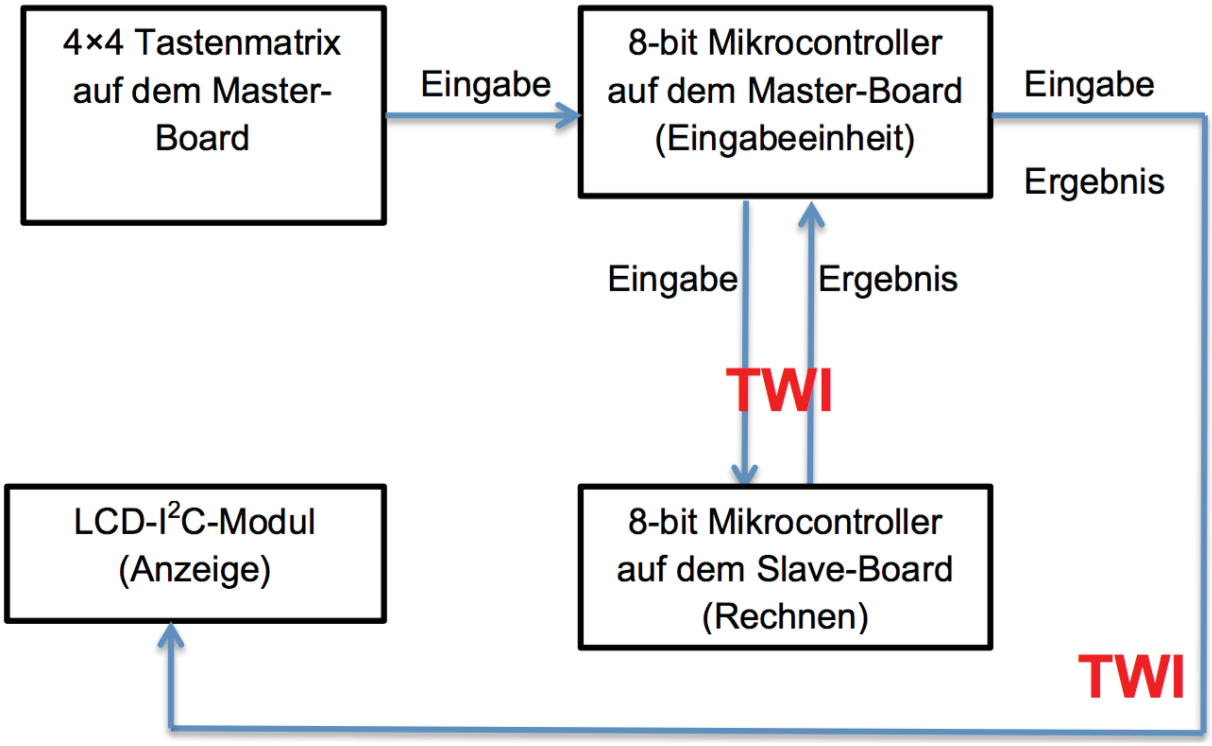

Abbildung 5: Entwurf eines wissenschaftlichen Taschenrechners

Die eingegebenen Symbole werden vom Master über den TWI an das LCD-Modul geschickt und außerdem in einem internen Cache gespeichert. Das LCD-Modul liest die Symbole und zeigt gleichzeitig auf dem LCD-Display an.

Wenn nach dem drücken der „=“-Taste das Ergebnis der Gleichung berechnet werden soll, schickt der Master durch den TWI-Bus die Gleichung im internen Cache an den Slave.

Nachdem der Slave die Berechnung beendet hat, wird das Ergebnis an den Master geschickt. Anschließend wird das Ergebnis als einen Operand für nächste Operation im internen Cache des Masters gespeichert. Zur gleichen Zeit sendet der Master dem LCD-Modul das Ergebnis, um es auf dem LCD anzuzeigen.

3.2 Kommunikation zwischen beiden Multifunktionsboard

Das Two-wire Serial Interface(TWI) ist für typische Mikrocontrollerapplikationen geeignet. Laut des TWI Protokolls ist es erlaubt, dass 128 unterschiedliche Geräte mittels zwei bidirektionaler Signalleitungen, Takt-(SCL) und Datenleitung(SDA), verbunden werden können. Und es benötigt auf jeder Signalleitung nur einen Pull-up Widerstand als externe Hardware. Jeder an den Bus angeschlossene Slave hat einzelne Adresse. [2]

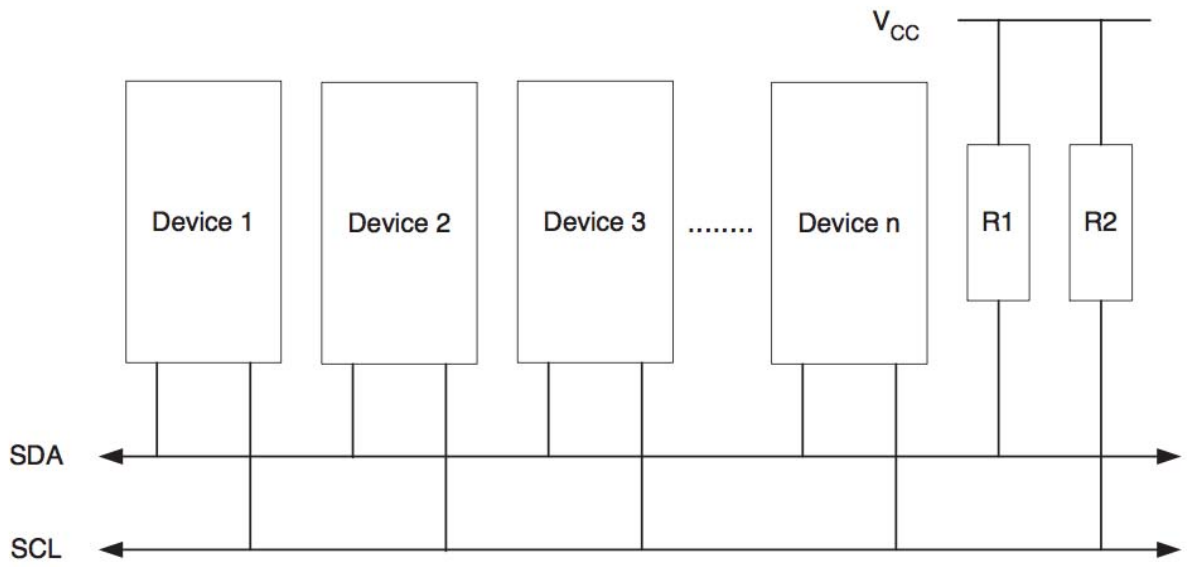

Abbildung 6: Die Zusammenschaltung des TWI[2]

Im Rahmen des TWI-Buses gibt es vier Rollen, Master, Slave, Sender sowie Empfänger. Der Master kontrolliert den Beginn und Ende einer Datenübertragung und erzeugt den Takt SCL. Der Slave wird mittels des Masters adressiert. Der Sender sendet Daten auf den Bus, die der Empfänger vom Bus liest.

Es gibt folgende wichtige Eigenschaften für das TWI:

- TWI unterstützt die Operationen des Masters und Slaves;
- die Devices können als Sender oder Empfänger arbeiten;

- TWI nutzt einen 7-bit Adressraum ermöglicht bis zu 128 verschiedene Slave-adressen;
- Datenübertragungsgeschwindigkeit ist bis zu 400 kHz;
- Die Slaveadressen und die Adresse der allgemeinen Aufrufe sind program-mierbar. [2]

Auf dem Mikrocontroller jedes Boards gibt es zwei der I/O Ports, PC0 und PC1, aus den TWI Busleitungen bestehen. PC0 arbeitet als den Takt SCL, und natürlich ist PC1 für Datenleitung SDA.

3.2.1 Datenübertragung des TWI-Buses

Der TWI-Bus überträgt fünf Datentypen, um eine Übertragung zu beginnen/beenden, Bits, Adresspaket und Datenpaket zu übertragen, sowie Kombination der Adress- und Datenpaket in einer Übertragung.

Übertragung des Bits

Die Übertragung jedes Datenbits auf dem TWI-Bus ist von einem Impuls auf der Taktleitung begleitet. Wenn die Taktleitung hoch ist, muss der Pegel der Datenleitung stabil sein, außer zur Erzeugung der Start- und Stoppbedingungen.[2]

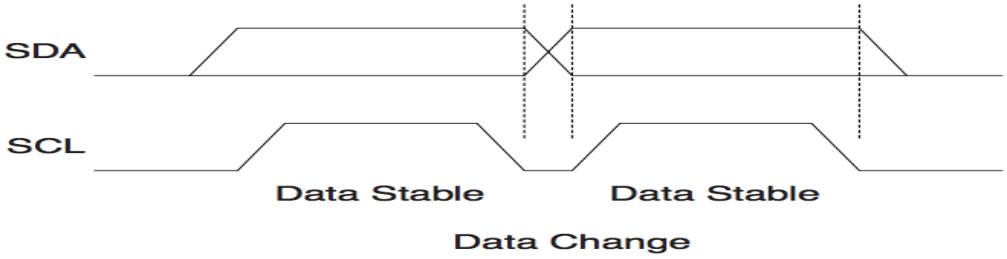

Abbildung 7: Datavalidität des TWI[2]

Start- und Stoppbedingung

Der Master beginnt und beendet eine Datenübertragung. Daher erzeugt der Master auf dem Bus eine Startbedingung, um die Datenübertragung zu beginnen. Und die Übertragung wird davon beendet, wenn der Master eine Stoppbedingung auf dem Bus sendet. [2]

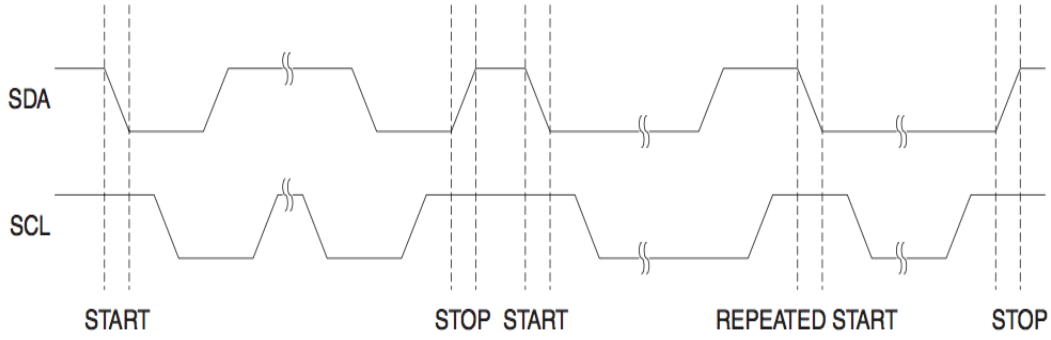

Abbildung 8: Start- und Stoppbedingung [2]

Adresspaket Format

Alle Adresspakete, die auf dem TWI Bus gesendet werden, sind von neun Bits Länge, die aus sieben Adressbits, einem READ/WRITE Bit und einem Bestätigungsbit(ACK). Falls der READ/WRITE Bit gesetzt ist, wird READ-Operation durchgeführt, sonst WRITE-Operation. Die Slave-adressen können frei vom Programmierer verteilt werden, aber die Adresse 0000 000 ist für den Rundruf reserviert. Alle Adressen, vom Format 1111 xxx, sind für eine zukünftige Benutzung reserviert. Das Adresspaket beinhält eine Slave-Adresse und ein READ oder WRITE Bit, das SLA+R oder SLA+W genannt wird. [2]

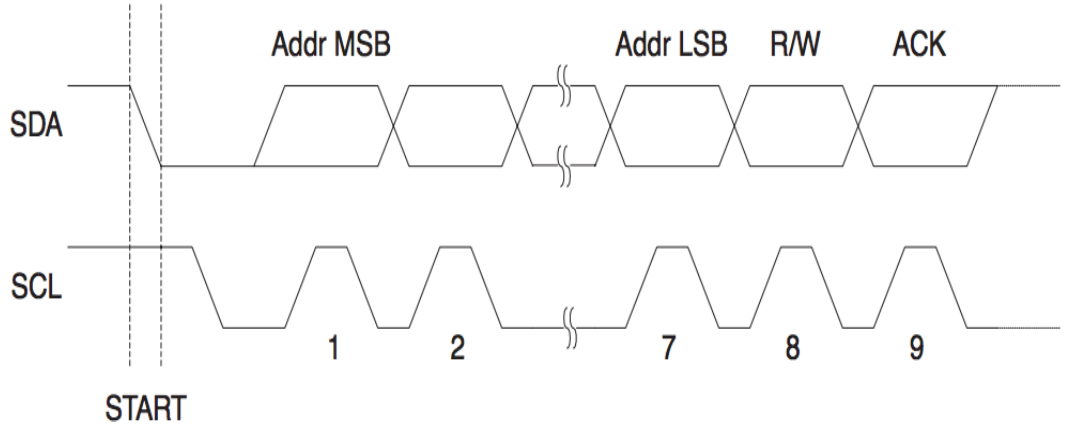

Abbildung 9: Adresspaket Format [2]

Datenpaket Format

Alle Datenpakete, die auf dem TWI Bus gesendet werden, sind von neun Bits Länge, bestehend aus einem Datenbyte und einem Bestätigungsbit(ACK). Während der Datenübertragung erzeugt der Master den Takt und die Start- und Stoppbedingung, die zur gleichen Zeit der Receiver verantwortlich empfängt. [2]

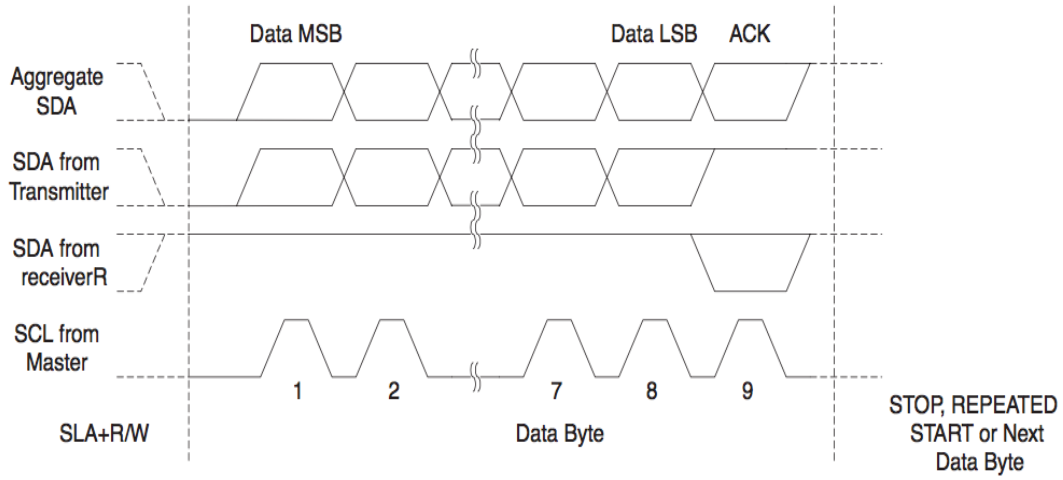

Abbildung 10: Datenpaket Format[2]

Kombination der Adress- und Datenpakete in die Datenübertragung

Eine Datenübertragung besteht aus einer Startbedingung, einem SLA+R/W, einer oder mehren Datenpaketen und einer Stoppbedingung. Eine Leermeldung, in der eine Stoppbedingung einer Startbedingung folgt, ist illegal. Und mit der Wired-ANDing der SCL Leitung kann es realisiert werden, dass Handshake zwischen Master und Slave durchgeführt wird. Dadurch, dass Slave die SCL Leitung auf Low-Pegel zieht, kann Slave die Low-Periode der SCL Leitung verlängert. Diese Eigenschaft ist dafür nützlich, wenn die errichtete Taktgeschwindigkeit des Masters für das Slave zu schnell ist, oder wenn das Slave mehr Zeit für die Verarbeitung der Datenübertragung benötigt. [2]

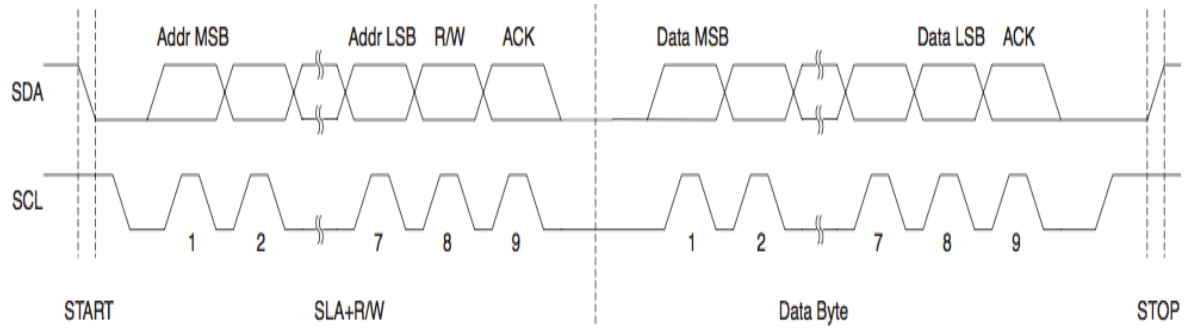

Abbildung 11: Typische Datenübertragung[2]

4 Ablaufplan und Programme

Vor der Programmierung sollte man zuerst die Ablaufpläne der Programme skizzieren. Der Programmablauf ist in ein Hauptprogramm und einige Unterprogramme eingeteilt. Unterprogramme arbeiten dafür, Variablen zu definieren, mehrfach Programmsegmente zu wiederverwenden, und Kommunikation zwischen beiden Multifunktionsboards zu herstellen, d.h. die Unterprogramme dienen dazu, Hauptprogramm zu vereinfachen. Folgende Ablaufpläne werden für einen wissenschaftlichen Taschenrechner erstellt.

4.1 Das Hauptprogramm(Master.c)

Auf dem Master-Board schreibt man das Hauptprogramm, in dem alle Funktionen des Masters implementiert werden. Weil der Master für Eingabe und Anzeige verantwortlich ist, ist die wichtige Aufgabe im Hauptprogramm „Master.c“ außer den Initialisierungen von Hardware und Systemen, zyklisch die Stände der Tasten zu entdecken. Es gibt vier multifunktionale Tasten, um acht Operatoren zu implementieren. Mit der Programmierung kann man nach falschem Druck einer multifunktionalen Taste nochmal gewünschte Taste drücken, damit die Operatoren wechseln. Alle Zustände der Tasten sind im Cache des Mikrocontrollers gespeichert. Nach dem Druck des Gleichheitszeichens bedeutet es, dass eine komplette arithmetische Gleichung fertig eingegeben wird. Nun sendet der Master dem Slave die Gleichung. Aufgrund des oben genannten Entwurfs wird der Ablaufplan des Hauptprogramms skizziert.

1. Hardware initialisieren: Vor der Programmierung auf den beiden Multifunktionsboards müssen die verwendeten Hardware initialisiert werden, nämlich I/O Ports, I^2C(TWI) und LCD. Die Initialisierung von I/O Ports enthält drei Teile: a) eine Komma, zehn Zahlen und 8 Operatoren auf sechzehn Tasten konfiguriert werden; b) Ports A und D als Eingang definiert werden; c) Port C0 und 1 als I^2C-Bus dienen. Die Initialisierung von I^2C enthält es, 7 Bit TWI Adresse, Übertragungsmodelle, und die Kommunikation von Master-Board und Slave-Board zu konfigurieren.

2. System zu initialisieren: dieser Schritt beinhält zwei Teile, den Cache der Tasten und Cache der Anzeige zu initialisieren.

3. Erste Entscheidung: Sind die Tasten gedrückt? Wenn nein, setzt sich diese Entscheidung fort. Wenn ja, läuft das Programm in nächsten Schritt. Bei der Polling Strategie kann man testen, ob eine Taste gedrückt wird. Polling Strategie ist eine

Methode, den Zustand der Ports häufig abzufragen, um die Zustandsänderung zu ermitteln. Mit drei Schritten kann man erkennen, ob die Taste gedrückt ist:

a) den Zustand der I/O Ports zu testen.

b) sich um 150ms zu verzögen. Das Ziel ist die Entprellung der Tasten.

c) den Zustand der I/O Ports nochmal zu testen. Wenn die beide Zustande gleich sind, hat man irgendeine Taste gedrückt.

4. Tastenverarbeitung: Welche Tasten werden gedrückt?
5. Zweite Entscheidung: ist die Taste „=“? Wenn nein, springt das Programm zum 3. Schritt. Wenn ja, läuft es in nächsten Schritt.
6. Nach dem Druck der Taste „=“ wird der Cache der Tasten dem Slave-Board gesendet.
7. Am Ende wartet das Master-Board auf die Ergebnisse, und danach werden die Ergebnisse auf LCD angezeigt.

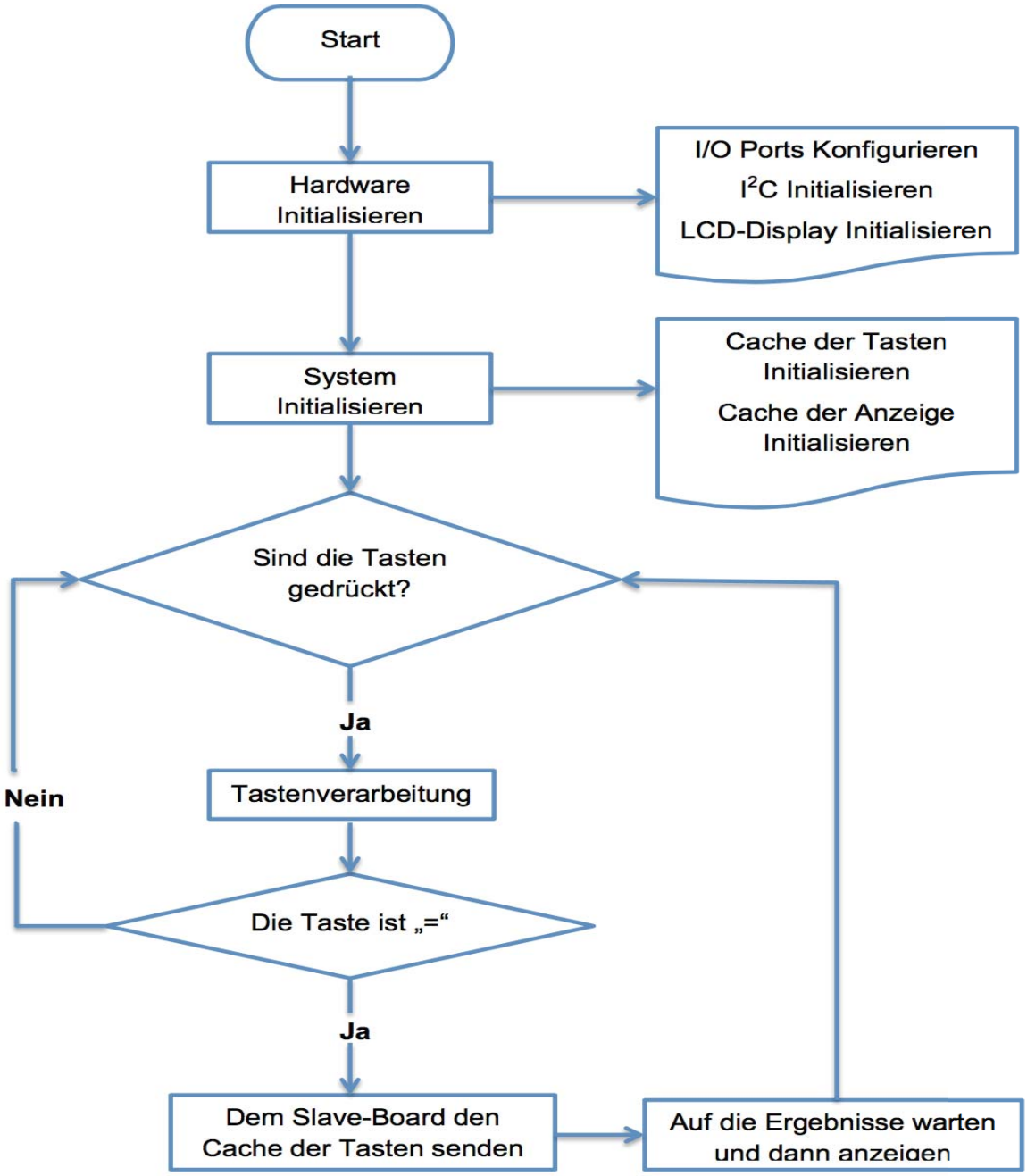

Abbildung 12: Ablaufplan des Hauptprogramms

4.2 Tastenverarbeitung

Wenn eine Taste sicher gedrückt war, soll der Computer anschließend im Keymap das Symbol der Taste ermitteln. Dieser Vorgang ist Tastenverarbeitung, die vier Entscheidungen beinhält. Die Funktion des Keymaps ist, dass I/O Ports in benutzerdefinierten Tasten abgebildet werden. Auf *Abbildung 13* wird es gezeigt.

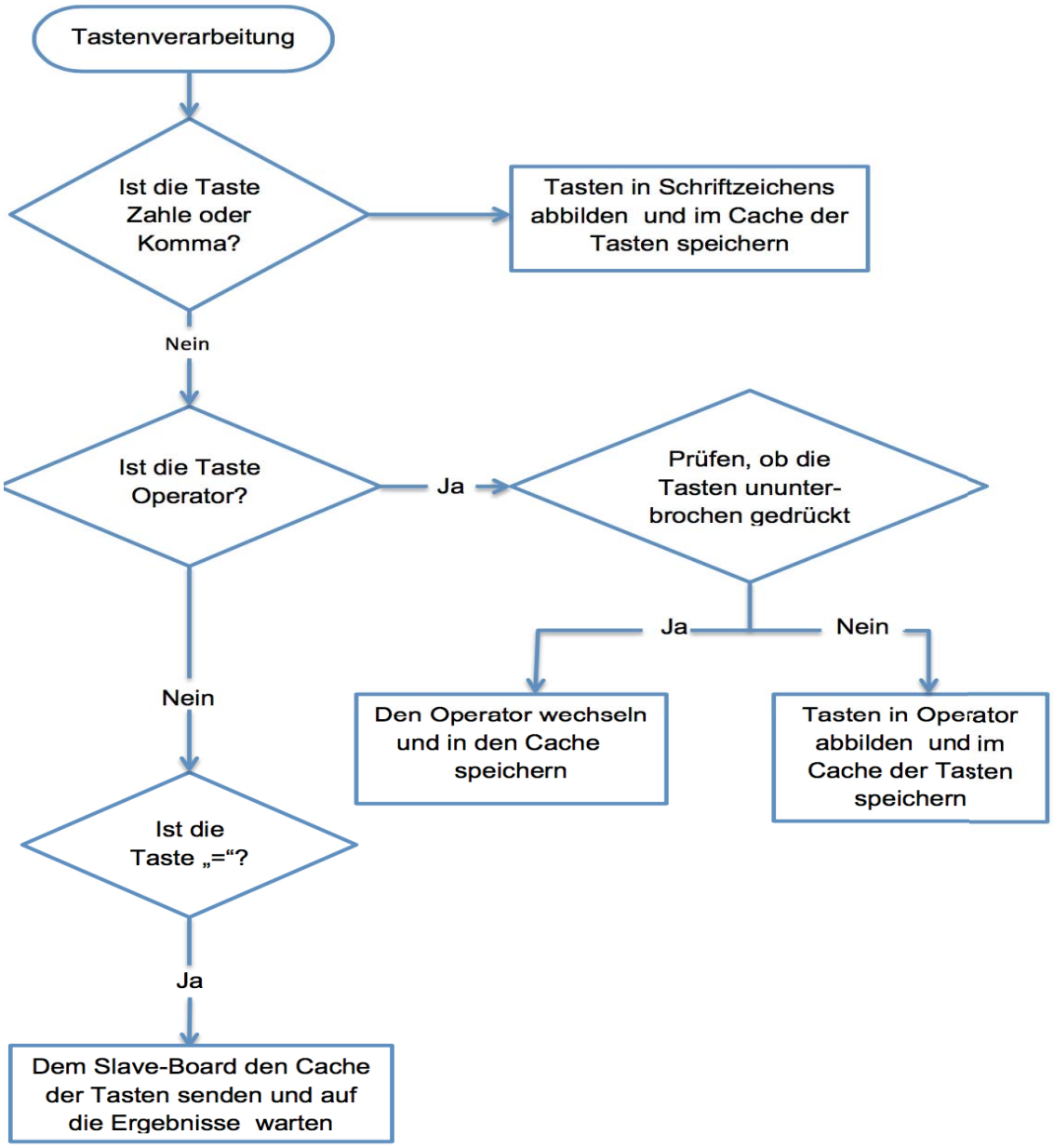

Abbildung 13: Ablaufplan der Tastenverarbeitung

1. Erste Entscheidung: ist die Taste Zahle oder Komma? Wenn nein, läuft das Programm weiter in die nächste Entscheidung. Wenn ja, werden die Zahlen oder Komma im Cache der Tasten gespeichert und gleichzeitig in LCD angezeigt. Weil nur ein Komma in einer Zahl existiert, wird zuerst ein Komma-Flag „dotctr“ gesetzt. Wenn die Taste des Kommas gedrückt wird, ist der Komma-Flag gesetzt. Es ist ungültig, dass man diese Taste mehrmals drückt.

2. Zweite Entscheidung: ist die Taste Operator? Wenn nein, setzt das Programm im nächsten Schritt fort. Wenn ja, geht das Problem in eine Verzweigung.

3. Falls die Taste ein Operator ist, kommt eine Verzweigung, ob die Taste ununterbrochen gedrückt wird. Denn es gibt auf vier Tasten 8 Operatoren, d.h. jede Taste hat zwei Funktionen, d.h. die Tasten der Operatoren sind multifunktionale Tasten. Wenn nein, wird die Taste im Operator abgebildet und im Cache der Tasten gespeichert und auf dem LCD angezeigt. Wenn ja, erzeugt es drei Situationen, deshalb muss der Mikrocontroller für geeignete Situation entscheiden:

a) Diesmalige multifunktionale Taste ist gleich der letzten Taste, d.h. die funktionalen Tasten drückt man mehrmals. Auf jedem Knopfdruck sollten sich die Operatoren ändern. Deshalb wenn man die vier multifunktionalen Taste mehrmals gedrückt, wechseln die Operatoren der Tasten ab.

b) Wenn nach einem Operator die Taste „minus“(„-“) gedrückt wird, bedeutet es, dass man für den zweiten Operator eine negative Zahl eingibt.

c) Sofern es obige zwei Situationen nicht entspricht, kommt die dritte Situation: dass eine multifunktionale Taste nach einer ganzen Gleichung gedrückt wird, gibt es keine Effekt auf LCD, denn in einer einfachen Gleichung ist der zweite Operator illegal. Im Programm ist jetzt Operatorsymbol „optsign“ gesetzt, und gleichzeitig endet die Tastverarbeitung.

4. Die letzte Entscheidung: ob diesmalige Taste „=“. Nach der Eingabe „=“ erkennt der µC auf dem Master-Board, ob die eingegebene Gleichung legal sind und es nur einen arithmetischen Operator in dieser Gleichung gibt. Wenn ja, dann sendet das Master-Board dem Slave-Board die Eingaben, welches auf die Ergebnisse wartet. Danach werden die Ergebnisse in LCD angezeigt. Im Programm kann die einfach genaue Gleitkommazahl mit der Funktion „snprintf“ in Zeichenketten umgewandelt werden, um in LCD anzuzeigen.

4.3 Das Programm des Slave-Boards

Das Slave-Board dient der Rechnung und Kommunikation. Die Arbeiten auf dem Slave-Board beinhalten 3 Schritte, die Daten zu empfangen, die Gleichung zu rechnen, und die Daten zurück zu senden. *Abbildung 14* zeigt den Ablaufplan auf dem Slave-Board.

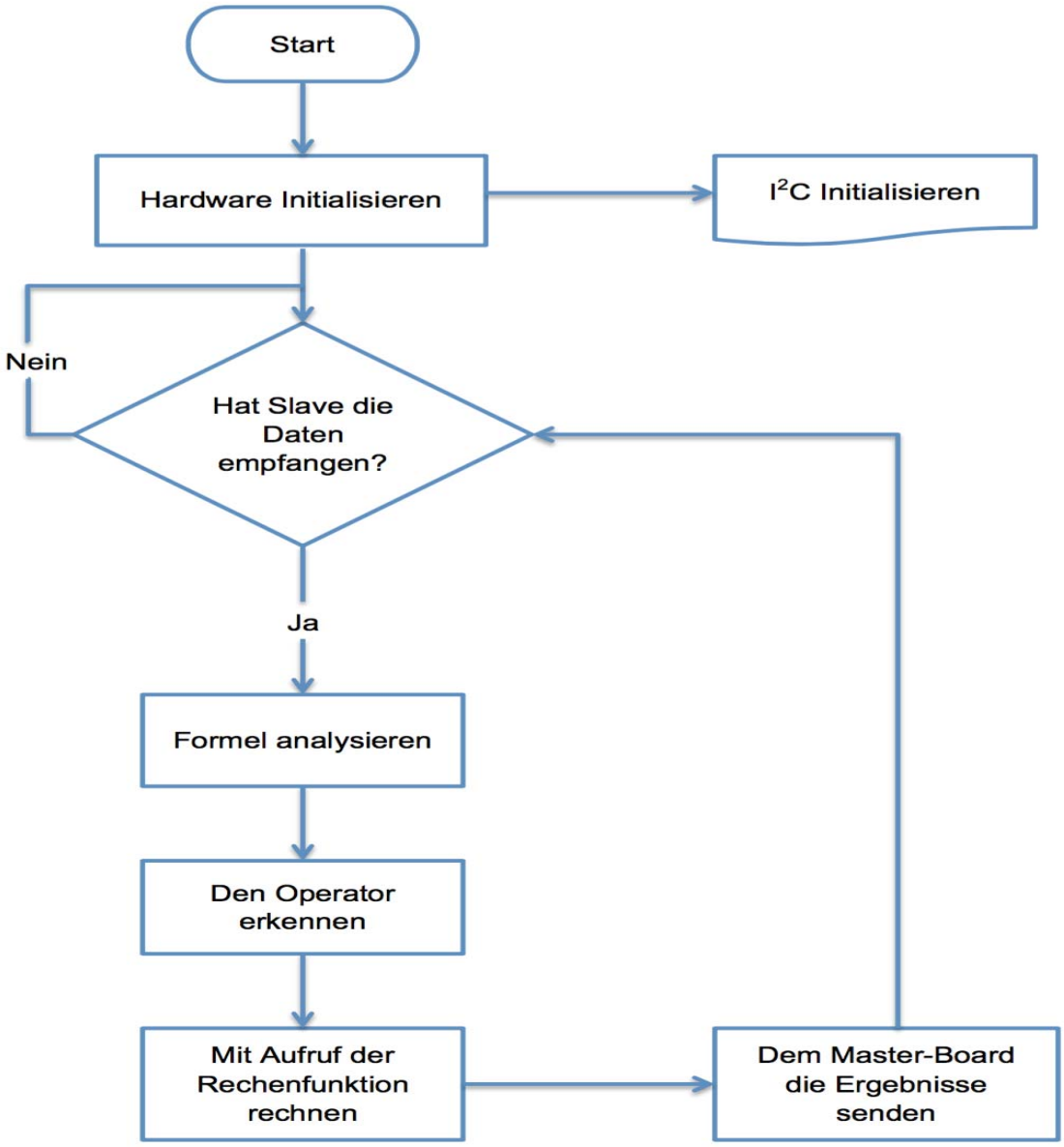

Abbildung 14: Ablaufplan des Slave-Board

1. Hardware initialisieren: I/O Ports(Port C0 und 1) werden zuerst konfiguriert, und gleichzeitig wird I^2C(TWI) als Slave Modus initialisiert. Der Unterschied vom Master-Board ist heri, dass im Statusregister SREG Bit7(I: Global Inerrupt Enable) und im TWI Control Register TWCR Bit0(TWIE: TWI Interrupt Enable) zugelassen wird.

2. Es gibt nur eine Entscheidung: ob das Slave-Board die Daten empfangen hat. Das Programm(Slave.c) enthält eine Endlosschleife(„while(1)“), um ständig zu entdecken, ob das Master-Board einige Daten gesendet hat. Wenn nein, setzt sich das Programm in der Endlosschleife fort. Wenn ja, dann läuft es in nächste

Tätigkeit. Im Programm wird ein Flag ’received’ eingestellt. Falls ‚received’ gesetzt ist, hat das Slave-Board die Daten vom Master schon erhalten. Und vor der Verarbeitung der Daten ist der Flag ‘received‘ im Programm gelöscht, denn die erhaltenden Daten werden anerkannt und das Slave-Board kann nächste Daten erhalten.

3. Formel analysieren: zum Senden der Daten wurden diese in eine Zeichenkette umgewandelt. Deshalb erhält das Slave-Board auch die Daten in den Zeichenketten. Um die Gleichung zu berechnen, müssen die Daten in Gleitkommazahl umgeformt werden. Zuerst kann der Operator einer Gleichung mittels der Funktion „strchr“ von den erhaltenden Zeichenketten gefunden werden. Danach weiß man, welche Operation das Slave-Board empfängt. Laut des Operators wird die Form der Operation erkannt, deshalb können die Zeichenkette vor und nach dem Operator in Gleitkommazahl um geformt werden. Anschließend wird ein komplett arithmetischer Ausdruck erstellt.

4. Den Operator erkennen: Nach der Analyse entdeckt µC auf dem Slave-Board negatives Symbol. Danach wird der Operator gefunden. Anschließend werden die beide Zahle einer Gleichung mittels der Position des Operator(eine Zahl vor dem Operator, andre nach dem Operator) in die Gleitpunktzahlen umgewandelt. Danach wird die Gleichung erstellt.

5. Mit Aufruf der Rechenfunktion rechnen: am Anfang des Programms werden ein Funktionszeiger und eine Struktur definiert, und anschließend wird eine Reihe der Funktionszeigern aufgestellt, damit die wörtlichen Operatoren zu den wirklichen Funktionen in den Operationen, z.B. ‚s’ in LCD bedeutet Sinus in einer wirklichen Gleichung. Am Ende des Programms bestehen acht definierte Gleichungen, mit den der erhaltende Ausdruck berechnet werden kann. Auf beiden Fällen sollen „error“ in den LCD gezeigt werden: a) wenn der Divisor gleich Null ist; b) wenn der Numerus des natürlichen Logarithmus’ kleiner als Null ist.

6. Am Ende sendet das Slave-Board dem Master-Board die Ergebnisse zurück.

4.4 ISR-Interrupt Service Routine

Wenn das Master-Board einige Daten sendet, musst µC auf dem Slave-Board zuerst seinen aktuellen Programmablauf unterbrechen und anschließend einen Interrupt behandeln. Nach der Interruptbehandlung kehrt µC in den vorherigen Programmab-

lauf zurück. Deshalb erzeugt die Kommunikation(durch das TWI-Bus) zwischen die beiden Multifunktionsboards einen Interrupt.

Es gibt auf dem Slave-Board zwei Übertragungsmodi, Slave Sender Modus und Slave Empfänger Modus. Bei den beiden Modi musst TWEN auf eins gesetzt werden, um TWI zu aktivieren. Zur gleichen Zeit wird TWEA(im TWCR Bit 6, TWI Enable Acknowledge) auch auf eins gesetzt, um die Bestätigungsnachricht ACK zurückzusenden.

1. Nachdem die Daten vom Master nach Slave gesendet geworden haben, wird die Endlosschleife des Slave unterbrochen und springt in das ISR Programm (in „i2cslave.c“), um diesen Interrupt zu verarbeiten.

2. Am Anfang wird geprüft, welcher Interrupt das Programm unterbricht. Der Slave Empfänger Modus besitzt neun verschiedene Interrupts, aber im Programm werden nur 3 Interrupts benutzt. Der Slave Sender Modus hat fünf Interrupt, wobei vom Programm wieder nur 3 genutzt werden. Alle Interrupts hat Anfang des Programms definiert und adressiert. Danach sind die Anweisungen „switch“ und „case“ benötigt. Nach der Schlüsselwort „switch“ wird ein Ausdruck geschrieben, in dem ein bitweises UND(„TWSR&0xF8“) existiert. Und nach der Schlüsselwort „case“ gibt es sechs verschiedene Interrupts in den beiden Übertragungsmodi. Damit die Typen der Interrupts erkannt werden, wird das Ergebnis des bitweisen UND mit den Adressen des Interrupts vergleicht.

3. Jeder der sechs Interrupts wird durch verschiedene Verfahren verarbeitet. Wenn der Interrupt SR_SLA_ACK(SLA+W von Slave hat empfangen geworden und eine Bestätigung „ACK“ wird zurückgesendet) ist, wird ISR unterbrochen und springt im Programm „slave.c“ zurück. Wenn nein, dann läuft das Programm weiter in nächstes Schlüsselwort „case“(SR_DATA_ACK). Ist das bitweisen UND gleich als 0x80? Wenn ja, empfängt das Slave-Board die Daten und ACK hat zurückgesendet. Danach werden die Daten in den Cache gespeichert. Natürlich endet das ISR-Programm und springt in das Slave-Programm. Wenn nein, dann setzt es sich in nächste Verzweigung fort, ob es ein Stopp- oder Restartsignal(SR_STOP) ist. Nach dem Empfang erhält das Slave-Board eine ganze Gleichung, deshalb kehrt es zum Slave-Programm zurück um die Gleichung zu berechnen.

4. Falls der Interrupt zu den oben genannten Fällen nicht gehört, entscheidet ISR, ob der Slave Sender Modus den Interrupt erzeugt. Erster Fall ist, ob das SLA+R vom Slave-Board (ST_SLA_ACK) empfangen wird und ACK zurückkommt. Wenn ja,

gibt es eine Entscheidung mit der Anweisung „if“, in der die Bedingung „ready“ existiert. „ready“ ist ein Flag und bedeutet, die Kalkulation zu erledigen. Deshalb wenn „ready“ auf eins gesetzt ist, wird das Ergebnis der Kalkulation zum Master-Board gesendet und dann wird der Flag gelöscht. Wenn der Flag nicht gesetzt ist, bricht ISR derzeitig ab und kehrt im Slave-Programm zurück. Zweiter Fall(ST_DATA_ACK) ist, ob Datenbyte im Register TWDR gesendet hat und das Slave-Board ACK erhält. Wenn ja, sendet das Slave-Board dem Master das Ergebnis der Kalkulation und ISR wird unterbrochen. Wenn nein, dann kommt der dritten Fall(ST_DATA_NACK): dass Datenbyte im Register TWDR wurde gesendet und erhiert eine Nichtbestätigung „NACK“ zurück, d.h. das Master-Board hat schon alle Daten erhalten oder kann nicht mehr die Daten empfangen.

5. Wenn alle sechs genannten Interrupts verneint werden, macht die Datenverarbeitung einen Fehler. Dann springt es im Slave-Programm.

Die *Abbildung 15* zeigt den Ablaufplan des ISR Programm.

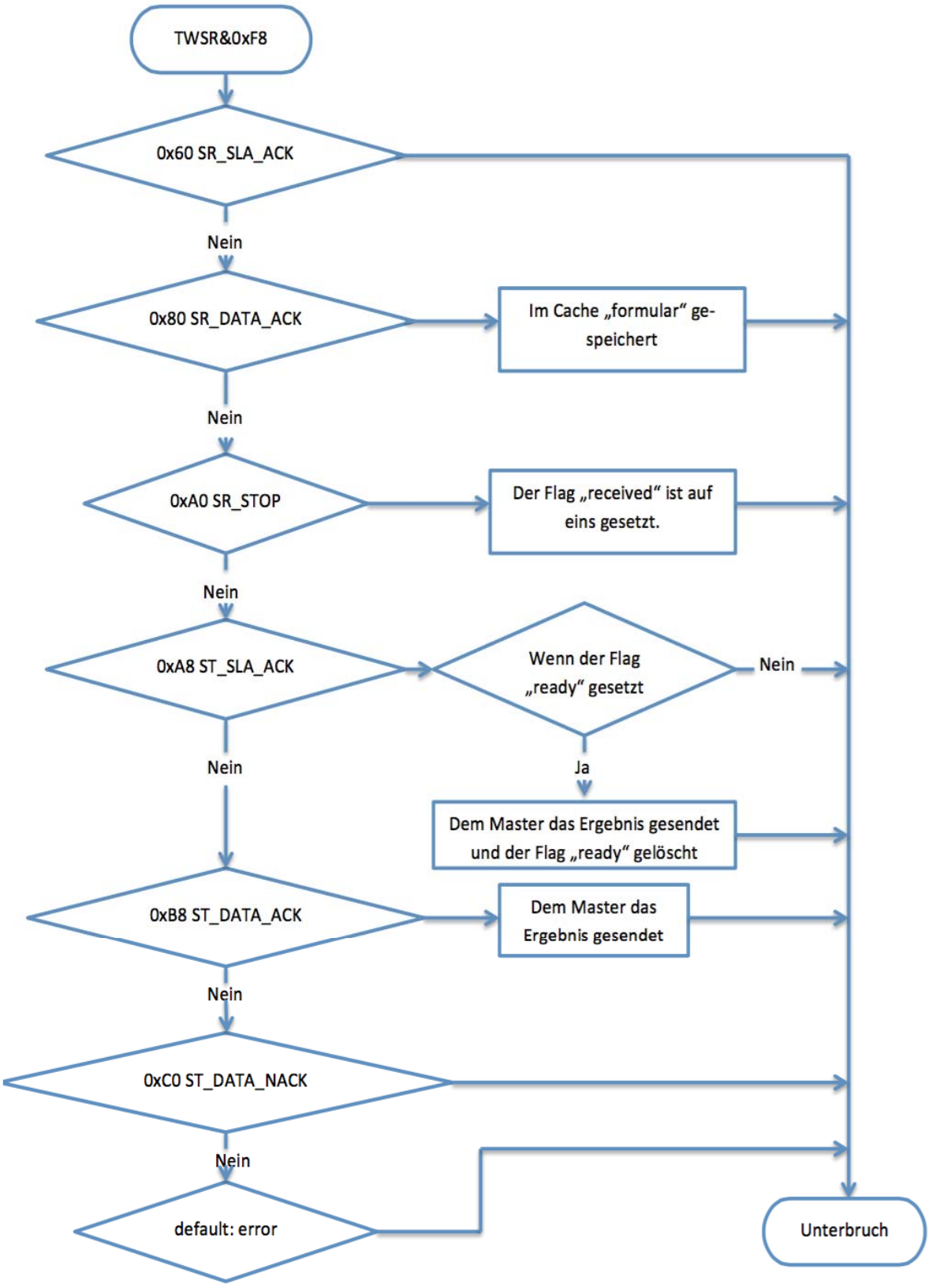

Abbildung 15: Ablaufplan des ISR Programms

5 Funktionstest

Nach der Verbindungen aller Hardware werden die Programme durch das JTAG-Emulator darauf kompiliert. Anschließend testet man, ob alle Funktionen vom Programm realisiert werden. Laut der Aufgabenstellung implementiert dieser wissenschaftliche Taschenrechner insgesamt acht arithmetische Funktionen, vier Grundrechenarten(add, sub, mul, div) sowie vier spezielle mathematische Funktionen (cos, sin, exp, ln). Während der Berechnung benutzen die Programme einfache Genauigkeit als Zahlen. Und in einer einfachen Genauigkeit gibt es nur sieben bis acht geltende Ziffern. Wenn über sieben Ziffern ins System eingegeben werden, kann die Eingabe auf dem LCD angezeigt werden, aber nach der Berechnung rundet der Mikrocontroller automatisch die achte oder neunte Ziffern des Ergebnisses, d.h. es können maximale sieben oder acht geltende Ziffern als Ergebnis auf dem LCD angezeigt werden. Vor dem Test wird der komplette Aufbau der Hardware in der *Abbildung 16* angezeigt. Im Test werden arithmetische Ausdrücke benutzt, um festzustellen ob diese mit dem Taschenrechner berechenbar sind.

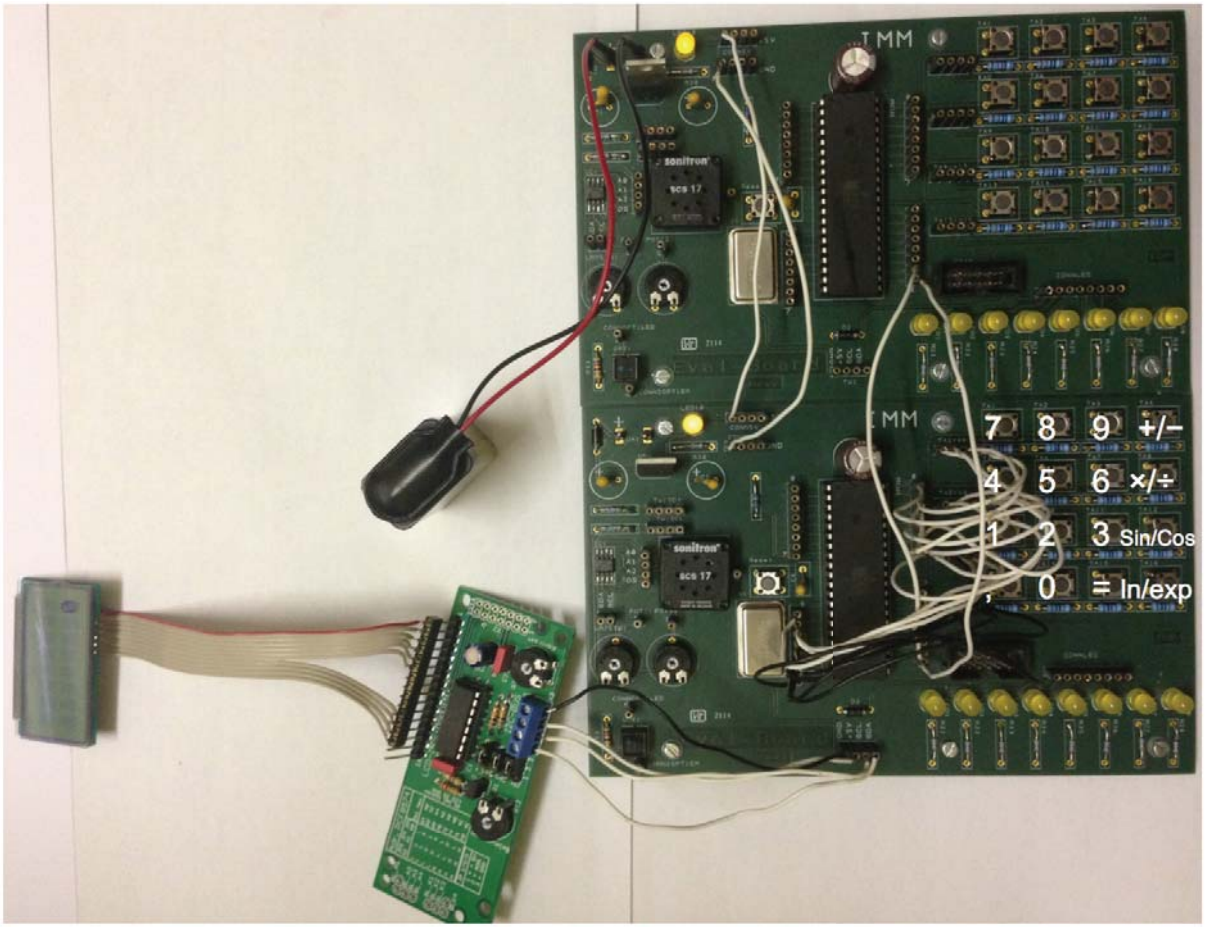

Abbildung 16: Kompletter Aufbau des Taschenrechners

5.1 Addition(a+b)

Addition ist eine Grundrechenart, deshalb sind die Berechnungen des Tests Additionen zwischen zwei mehrstellige Ganzzahlen oder Dezimalen, inklusive negativer Zahlen. In der *Tabelle 1* werden Ausdrücke und seine Ergebnisse gezeigt. Um „+“ einzugeben, drückt man einmal TA4 auf der Tastenmatrix.

Tabelle 1: Addition(a+b)

Ausdrücke	Ergebnis	Berechenbar?
9 758 634+1 343 564	1,110 22e+07	Ja
9 866 999 999 999+54	9,867e+12	Ja
47 512,125+5 142,4	52 654,5	Ja
-5 896 426+6 542 899	646 473	Ja

Weil alle Operanden einfache Genauigkeit sind, benutzt der Mikrocontroller wissenschaftliche Notation, wenn der Wert das Ergebnis zu groß ist. „e+07“ bedeutet 10 hoch 7.

5.2 Subtraktion(a−b)

Die Berechnungen des Tests sind Subtraktionen zwischen zwei mehrstellige Ganzzahlen oder Dezimalen, inklusive negativer Zahlen als Minuenden. In der *Tabelle 2* werden Ausdrücke und seine Ergebnisse gezeigt. Um „−“ einzugeben, drückt man zweimal TA4 auf der Tastenmatrix.

Tabelle 2: Subtraktion(a−b)

Ausdrücke	Ergebnis	Berechenbar?
89 564 257−6 684 321	8,287 99e+07	Ja
5 934 256−65 242 365	-5,930 81e+07	Ja
-243,147−651,724	-894,871	Ja
-52−75 823 143 589	-7,582 31e+10	Ja

5.3 Multiplikation(a×b)

Multiplikation ist ein bisschen komplizierter als Addition und Subtraktion, weil die Begrenzung der Ergebnisse größer ist. In der folgenden Tabelle werden zwei mehrstellige Ganzzahlen oder Dezimalen, oder sehr kleine Zahlen, und noch mit negativer Zahlen multipliziert, um die Berechenbarkeit der Ausdrücke zu entdecken. In der *Tabelle 3* werden Ausdrücke und seine Ergebnisse gezeigt. Um „×“ einzugeben, drückt man einmal TA8 auf der Tastenmatrix.

Tabelle 3: Multiplikation(a×b)

Ausdrücke	Ergebnis	Berechenbar?
59 531 465×2 546 824	1,516 16e+14	Ja
485 613 256×(-53 654)	-2,60551e+13	Ja
0,1253×0,021	0,0026313	Ja
5 599 216 423×(-0,12)	-6,71906e+8	Ja

5.4 Division(a÷b)

Der letzte Test der Grundrechenarten ist Division, mit der man den Quotienten bekommt. In der *Tabelle 4* werden Ausdrücke und seine Ergebnisse gezeigt. Um „÷“ einzugeben, drückt man zweimal TA8 auf der Tastenmatrix.

Tabelle 4: Division(a÷b)

Ausdrücke	Ergebnis	Berechenbar?
587÷3	195,667	Ja
1÷546 82	1,82876e-05	Ja
425÷(-54,5323)	-7,79355	Ja
1 248 652÷45 635 698	0,0273613	Ja

50 423÷0	error	Nein
19 250 427÷(-0,1991)	-9,66872e+07	Ja

Auf dem LCD wird einfache Genauigkeit als Ergebnis benutzt. Obwohl der Quotienten im ersten Ausdruck eine periodische Dezimale, rundet der Mikrocontroller den Quotienten automatisch vor der Anzeige auf. Im fünften Ausdruck ist der Divisor gleich Null und nicht berechenbar, deshalb zeigt LCD „error“. Wenn Divisor null ist, ist der Ausdruck mit diesem Taschenrechner nicht berechenbar.

5.5 Sinusfunktion(Sin(a))

Mit den vier speziellen mathematischen Funktionen wird einen wissenschaftlichen Taschenrechner implementiert. In den Programmen wird „math.h“ aufgerufen, um mathematische Funktionen zu verwenden, und „math.h“ arbeitet mit Radiant, der die Einheit des Winkels ist. Die Sinusfunktion wird mit sehr großer oder kleiner Zahlen, und einige besondere Zahlen, wie z.B. halb Pi, Null und eineinhalb Pi berechnet. Und *Tabelle 5* zeigt den Vorgang dieses Tests. Um „sin“ einzugeben, drückt man einmal TA12 auf der Tastenmatrix.

Tabelle 5: Sinusfunktion(Sin(a))

Ausdrücke	Ergebnis	Berechenbar?
Sin(512 356)	-0,0626278	Ja
Sin(999 999 999 999 999)	-0,699446	Ja
Sin(0,000 000 001)	1e-09	Ja
Sin(0)	0	Ja
Sin(1,570795)	1	Ja
Sin(4,712388)	-1	Ja

Mit irgendeinem großen oder kleinen Winkel ist das Ergebnis der Sinusfunktion von minus eins bis zu eins.

5.6 Kosinus-Funktion(cos(a))

Im Test wird die Kosinus-Funktion mit großen, kleinen und besonderen Zahlen, wie z.B. null und Pi berechnet. Es kann die Begrenzung der Kosinus-Funktion bestätigen. Und *Tabelle 6* zeigt den Test der Kosinus-Funktion. Um „cos“ einzugeben, drückt man zweimal TA12 auf der Tastenmatrix.

Tabelle 6: Kosinus-Funktion(cos(a))

Ausdrücke	Ergebnis	Berechenbar?
Cos(999 999 999 999 999)	-0,714685	Ja
Cos(0,000 000 000 000 1)	1	Ja
Cos(-4139)	-0,0483021	Ja
Cos(-9,99)	-0,84447	Ja
Cos(0)	1	Ja
Cos(3,14159)	-1	Ja

Und die Größe der Kosinus-Funktion ist immer von minus eins bis zu eins. Und alle Winkel arbeiten mit Radiant. Auf jeden Fall kann die Kosinus-Funktion berechnet werden.

5.7 Exponentialfunktion(e^a)

Der Test der Exponentialfunktion ist dafür, welchen Zahlen und wie groß es mit diesem Taschenrechner nicht berechnen kann. Und die Basis der Exponentialfunktion ist Eulersche Zahl „e“. *Tabelle 7* zeigt den Vorgang des Tests. Um „exp“ einzugeben, drückt man einmal TA16 auf der Tastenmatrix.

Tabelle 7: Exponentialfunktion(e^a)

Ausdrücke	Ergebnis	Berechenbar?
$e^{88,8}$	inf	nein

$e^{88,7}$	3,325 98e+38	Ja
e^{0}	1	Ja
$e^{-88,7}$	3,00663e-39	Ja
$e^{-999\ 999\ 999\ 999\ 99}$	0	Ja
e^{1}	2,71828	Ja

Wenn die Eingabe größer als 88,7, kann dieser Taschenrechner das Ergebnis nicht berechnen. Deshalb zeigt es auf dem LCD „inf“, und es bedeutet für den Taschenrechner, dass das Ergebnis den darstellbaren Zahlenbereich überschreitet. Außerdem je kleiner der Exponent ist, desto mehr nähert sich der Wert Null, d.h. der Wert ist immer größer als Null.

5.8 Natürlicher Logarithmus(ln x)

Die Exponentialfunktion ist inverse Funktion vom natürlichen Logarithmus. Wegen der Eigenschaften des Logarithmus werden die Ausdrücke mit negativer Zahl und null besonders getestet. Im *Tabelle 8* wird das Ergebnis der Tests gezeigt. Um „ln“ einzugeben, drückt man zweimal TA16 auf der Tastenmatrix.

Tabelle 8: Natürlicher Logarithmus(ln x)

Ausdrücke	Ergebnis	Berechenbar?
ln(999 999 999 999 999)	34,5388	Ja
ln(25346,2)	10,1404	Ja
ln(-5)	error	nein
ln(0)	-inf	nein
ln(1)	0	Ja

Wegen der inversen Funktion von der Exponentialfunktion ist „x“ gleich als „e^{a}“. Daher wenn „x“ kleiner als Null ist, ist der Ausdruck falsch und gleichzeitig zeigt es auf

dem LCD „error“. Wenn „x“ gleich Null ist, wird der Wert des Logarithmus gegen minus Unendlichkeit konvergieren, deshalb zeigt es „-inf“.

5.9 Fortsetzende Berechnungen

Wenn dieser Taschenrechner eine Berechnung beendet, speichert der µC das Ergebnis. Anschließend kann dies Ergebnis als ein Operand verwendet werden, d.h. man kann mit dem andere Gleichung erstellen. In der *Tabelle 9* wird der Vorgang gezeigt.

Tabelle 9: Fortsetzende Berechnungen

Anzahl der Berechnung	Welche Tasten gedrückt?	Gleichung	Ergebnis
1.	1 6 + 9 2	16+92	108
2.	× 2 3	108×23	2484
3.	− 5 2 8 0 , 5	2 484−5 280,5	-2796,5
4.	÷ - 2 0	-2796,5÷(-20)	139,825
5.	l	ln(139,825)	4,94039
6.	e	$e^{4,94039}$	139,825
7.	s	Sin(139,825)	0,999709
8.	c	Cos(0,999709)	0,540547
9.	l	ln(0,540547)	-0,615174

„l“ bedeutet ln. „s“ ist Sin. „c“ ist für Cos. „e“ bedeutet exp. Die vier sind im Programm definiert, um auf dem LCD einfach anzuzeigen.

6 Zusammenfassung

Die Bachelorarbeit realisiert vier Grundrechenarten (add, sub, mul, div) sowie vier spezielle mathematische Funktionen(cos, sin, exp, ln) eines wissenschaftlichen Taschenrechners. Durch die Programmierung in *„AVR Studio 4.0“* auf zwei Mikrocontrollern „ATmega32“ ist es erledigt, Operation auf dem Master-Board einzugeben, die Eingabe auf dem Slave-Board zu berechnen und die Kommunikation mittels des TWI-Buses zwischen beiden Boards. Außer den Anforderungen der Bachelorarbeit kann dieser wissenschaftliche Taschenrechner fortsetzende Berechnungen machen. Und Slave-Board entdeckt Berechenbarkeit der Eingaben. Wenn es eine falsche oder übergroße Ausdrücke einzugeben, kann „error“ und „inf“ auf dem LCD angezeigt werden. Zur gleichen Zeit kann man die Operatoren freiwillig wechseln, um die gewünschten Operator zu wählen.

Obwohl die Bachelorarbeit fortsetzend berechnen kann, ist es nicht möglich mehrere Operanden in einer Eingabe zu verwenden. Deshalb kann diese Funktion in Zukunft entwickelt werden. Das Verbinden im Experimentieraufbau mit Steckkabeln ist einfach, kann aber zu Kurzschluss oder Trennung führen. Löten oder Flachbandkabel sind besser als Leitung für die Verbindung.

Literaturverzeichnis

[1] Prof. Dr. –Ing. Alexander Lampe. Bachelorarbeit „Programmierung eines wissenschaftlichen Taschenrechners auf zwei Multifunktionsboards“

[2] ATmega32/ATmega32(L) Datasheet (http://www.atmel.com/images/doc2503.pdf)

[3] Prof. Dr. –Ing. O. Hagenbruch Die Handbücher zum Programmiermodell des AVR ATmega128

Danksagung

Letzt möchte ich allen Personen danken, die mich bei der Erstellung meiner Bachelorarbeit unterstützt haben.

Erstens möchte ich mich bei meinen zwei Prüfer Prof. Dr. –Ing. Alexander Lampe und Herrn M. Sc. Süß Markus bedanken.

Weiteren danke ich Herrn Martin Müller und Herrn Christian Thormann, die mir durch Ihr Fachwissen geholfen haben.

Anlagen

1. Quelltext des Programms(auf CD-Rom)

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum Vorname Nachname